

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of the Claims:

1. (Currently Amended) A computer-implemented method for executing computer code by increasing the performance of a virtual machine that uses an interpreter to execute virtual machine program instructions, the computer-implemented method comprising:

obtaining, by an interpreter, a virtual machine program instruction to be executed by the virtual machine;

determining, by the interpreter, whether the virtual machine program instruction is a branch instruction;

determining, by the interpreter, whether a basic block is present in a code cache that stores native code corresponding to virtual machine program instructions when it is determined that the virtual machine program instruction is a branch instruction to be executed by said virtual machine, wherein the basic block is associated with a case block of the interpreter, wherein the case block corresponds to ~~associated with~~ the virtual machine program instruction and the basic block would include ~~includes~~ native code corresponding to the virtual machine program instruction ~~that has been previously interpreted and executed by the interpreter~~, thereby determining by the interpreter whether native code needed to execute the virtual machine program instructions is available in the code cache ~~being associated with the virtual machine~~;

~~executing, by the interpreter, the~~ native code included in the basic block present in from said the code cache when it is determined by the interpreter that the basic block is present in the code cache and the virtual machine program instruction is a branch instruction;

interpreting, by the interpreter, the virtual machine instruction when said determining determines that the basic block associated with the case block of the interpreter is not present in ~~said the~~ code cache, wherein said interpreting generates native code for the virtual machine instruction; and

copying the native code generated by the interpreter into ~~said cache the~~ code cache after the interpreting of the virtual machine program instruction ~~the code~~ by the

interpreter when said determining determines that the basic block associated with the case block of the interpreter is not present in ~~said the~~ code cache so that subsequently the native code can be obtained from the code cache, thereby enhancing the performance of the virtual machine.

2. (Canceled)

3. (Currently Amended) A computer-implemented method as recited in claim [[2]] 1 further including:

allocating space in the code cache for the code corresponding to the program instruction; and

providing the code corresponding to the program instruction with a label.

4. (Original) A computer-implemented method as recited in claim 3 further including placing the label in a table of labels.

5. (Currently Amended) A computer-implemented method as recited in claim [[2]] 1 wherein determining whether the basic block is present in the code cache includes searching through a table of labels to determine if a target associated with the program instruction has a matching label in the table of labels.

6. (Currently Amended) A computer-implemented method as recited in claim [[2]] 1 wherein the program instruction is a bytecode, and wherein the bytecode is executed by an interpreter of the virtual machine.

7. (Currently Amended) A computer-implemented method as recited in claim [[2]] 1 wherein the code cache is a native code cache, and the code corresponding to the program instruction is native code.

8. (Original) A computer-implemented method as recited in claim 1 wherein the program instruction is a bytecode and the code cache is a native code cache.

9. (Previously Presented) A computer-implemented method as recited in claim 8 further including interpreting the bytecode when the determining determines that the program instruction is not a branch instruction.

10. (Previously Presented) A computer-implemented method as recited in claim 1 further including:

computing a target using the program instruction, wherein the determining of whether the basic block is present in the code cache includes determining if the code cache includes any basic blocks which correspond to the target.

11- 27 (Canceled)

28. (Currently Amended) A computer system for increasing the performance of a virtual machine that uses an interpreter to execute virtual machine program instructions, wherein the computer system is operable ~~capable of operating~~ to:

obtain a virtual machine program instruction to be executed by the virtual machine;

determine whether the virtual machine program instruction is a branch instruction;

determine whether a basic block is present in a code cache that stores native code corresponding to virtual machine program instructions when it is determined that the virtual machine program instruction is a branch instruction to be executed by said virtual machine, wherein the basic block is associated with a case block of the interpreter, wherein the case block corresponds to ~~associated with~~ the virtual machine program instruction and the basic block would include ~~includes~~ native code corresponding to the virtual machine program instruction ~~that has been previously interpreted and executed by the interpreter~~, thereby determining by the interpreter whether native code needed to execute the virtual machine program instructions is available in the code cache;

execute the native code included in the basic block present in ~~from said the~~ code cache when it is determined by the interpreter that the basic block is present in the code cache and the virtual machine program instruction is a branch instruction;

interpret the virtual machine instruction when said determining determines that the basic block associated with the case block of the interpreter is not present in ~~said~~

the code cache, wherein said interpreting generates native code for the virtual machine instruction; and

copy the native code generated by the interpreter into the ~~cache~~ code cache after the interpreting of the virtual machine program instruction ~~the code~~ by the interpreter when said determining determines that the basic block associated with the case block of the interpreter is not present in ~~said~~ the code cache so that subsequently the native code can be obtained from the code cache, thereby enhancing the performance of the virtual machine.

29. (Currently Amended) A computer readable medium including computer program code for increasing the performance of a virtual machine that uses an interpreter to execute virtual machine program instructions, wherein computer readable medium comprises:

computer program code for obtaining by an interpreter a virtual machine program instruction to be executed by the virtual machine;

computer program code for determining by the interpreter whether the virtual machine program instruction is a branch instruction;

computer program code for determining by the interpreter whether a basic block is present in a code cache that stores native code corresponding to virtual machine program instructions when it is determined that the virtual machine program instruction is a branch instruction to be executed by said virtual machine, wherein the basic block is associated with a case block of the interpreter, wherein the case block corresponds to ~~associated with~~ the virtual machine program instruction and the basic block would include ~~includes~~ native code corresponding to the virtual machine program instruction that has been previously interpreted and executed by the interpreter, thereby determining by the interpreter whether native code needed to execute the virtual machine program instructions is available in the code cache ~~being associated with the virtual machine;~~

computer program code for executing ~~by the interpreter~~ the native code included in the basic block present in ~~from said~~ the code cache when it is determined by the interpreter that the basic block is present in the code cache and the virtual machine program instruction is a branch instruction;

computer program code for interpreting by the interpreter the virtual machine instruction when said determining determines that the basic block associated with the

case block of the interpreter is not present in ~~said~~ the code cache, wherein said interpreting generates native code for the virtual machine instruction; and

computer program code for copying the native code generated by the interpreter into the ~~cache~~ code cache after the interpreting of the virtual machine program instruction ~~the code~~ by the interpreter when said determining determines that the basic block associated with the case block of the interpreter is not present in ~~said~~ the code cache so that subsequently the native code can be obtained from the code cache, thereby enhancing the performance of the virtual machine.